

FIRST ASSIGNMENT

Group number : *Acme-30*

Student Names and Numbers

Emanuele Santo Iaia 1924549

Simone Giordano 1772347



SAPIENZA
UNIVERSITÀ DI ROMA

Initial Brainstorming	1
Setup Of Infrastructure For Ipv6 Addressing	1
Host Configuration	2
Problem Fixed	2
Dns Configuration	3
IPv4	3
IPv6	3
Host	4
Evaluation Of The Security Policy	5
Services of the ACME co.	5
Log File	5
Proxy	6
Secure Shell	6
Policy Implementation In Opnsense	6
MAIN FIREWALL	6
DMZ	6
EXTERNAL_CLIENTS	7
INTERNAL	7
WAN	7
INTERNAL FIREWALL	7
CLIENT	7
EXTERNAL	8
SERVERS	8
Test Of The Configuration	8
Firewall	8
DNS	8
SSH	9
Web Server	9
Log File	9
Final remarks	10

Initial Brainstorming

The first phase was to configure properly the firewall of the main router in order to make it able to connect the opensense webgui of the 2 routers from the host machine (our machine).

For the IPv6 addressing we needed to be able to get the prefix delegation from the ISP and then assign the correct IPv6 addresses to the host of the 2 network

After doing that, we moved on configuring the DNS server in order to correctly resolve the address of the entire network.

At the end we managed the firewall rules configuring all the services that the network had to serve.

Setup Of Infrastructure For Ipv6 Addressing

As mentioned before in this section we have to be able to get the correct prefix delegation from the isp. Knowing that the main firewall was connected to the isp through the wan interface and later we enabled the DHCPv6 configuration requesting only IPv6 prefix with the /56 Prefix delegation size.

To properly configure the host IPv6 address assignment we moved on each interface heading to the host (DMZ, EXTERNAL, INTERNAL) and we tracked the interface based on prefix delegation of the wan. Each interface had a different IPv6 prefix ID:

DMZ	0x0
INTERNAL	0x1
EXTERNAL_CLIENT	0x2

Regarding the internal firewall we repeated the same steps as before but the external interface works as the wan interface does in the main firewall, this means that we had to enable the DHCPv6 configuration on it and track the IPv6 addressing on the other interfaces based on the external one:

SERVERS	0x0
CLIENT	0x1

Host Configuration

to properly configure the host to be able to create an IPv6 address based on the router advertisement sent by the routers (EXTERNAL FIREWALL AND INTERNAL FIREWALL) we modified the sysctl.conf file located in the /etc/ folder of each host in such a way that were enabled to manage IPv6 addresses (net.ipv6.conf.all.disable_ipv6=0) then we added the rules to manage the following: the host behaving as an host disabling the forwarding of packets, accepting routers advertisement and router solicitation. Considering that the ip addresses couldn't be random we set ADDR_GEN_MODE=0 generating an IPv6 address based on EUI64 (mac addr based).

```
net.ipv6.conf.all.forwarding = 0
```

```
net.ipv6.conf.all.accept_ra = 1
```

```
net.ipv6.conf.all.addr_gen_mode = 0
```

```
net.ipv6.conf.all.use_tempaddr = -1 (optional to make the ipv6 address no temporary)
```

Problem Fixed

During the test phase we occurred in a ping problem in which cannot ping from any host of internal network since the echo reply wasn't correctly forwarded by the internal router to the source host. This problem was caused by an error in the IPv6 gateway. We solved the problem disabling the dynamic gateway and so we let the forwarding only on the fixed routes saved in the routers.

Dns Configuration

Regarding the DNS configuration we choose to follow 2 paths, with respect to IPv4 and IPv6 addressing.

IPv4

We followed the instruction to configure the DNS zentyal DNS server located in the Domain controller Host, in such a way the `/etc/bind/db.giordele.com` was created containing the pairings between the name of the host and his IPv4 ADDRESS.

IPv6

Considered the fact that zentyal does not provide the DNS RESOLVER for IPv6 , we choose to use UNBOUND DNS service provided by the two routers (main firewall and internal firewall). Doing so we override the giordele.com domain (IPv4) to send the requests to the Domain Controller

Domain Overrides		
Domain	IP	Description
giordele.com	100.100.1.2	DNS IPV4

instead of the giordele6.com Domain (IPv6) where we configured the STATIC RESOLVERS to provide a correspondence between the IPv6 address and the name of the host

Host	Domain	Type	Value	Description
aw	giordele6.com	AAAA	2001:470:b5b8:1ef1:b89d:d3ff:feec:ea07	AW
ce	giordele6.com	AAAA	2001:470:b5b8:1ef1:47bf:93d2:3073:7049	CE
dc	giordele6.com	AAAA	2001:470:b5b8:1ef0:4c34:16ff:fe3d:beb3	DC
ls	giordele6.com	AAAA	2001:470:b5b8:1ef0:14cd:d6ff:fe00:4e4c	LS
pc	giordele6.com	AAAA	2001:470:b5b8:1e02:e480:50ff:fe76:1546	PC
ps	giordele6.com	AAAA	2001:470:b5b8:1e00:8410:9bff:fe35:525c	PS
ws	giordele6.com	AAAA	2001:470:b5b8:1e00:40fa:57ff:fe4a:2073	WS

Host

In the hosts we considered 2 approaches depending on the IPv4 type of address assigned:

- *STATIC ADDRESSING: (pc.100, pc.254.ARPWATCH)*

We modified the interface file contained in the /etc/network/interfaces adding the rule of the “dns-resolver<Internal Firewall interface IPv4 address>”

- *DYNAMIC ADDRESSING: (CLIENT.ex1)*

We modified the interface file contained in the /etc/network/interface adding the rule “auto inet dhcp”. Moreover we modified /etc/dhcp/dhclient.conf file adding the rule “pretend domain-name-servers <Main Firewall interface IPv4 address>”

Evaluation Of The Security Policy

We set up the interfaces of the two routers heading to each other in a way to accept all the traffic between them in order to insert the reject rules directly on the firewall's rules of the network.

The "Client Ex 1" host of the "**External Network**", can only access the Internet, the page of the webserver and no other services, allowing just the services in the ports 53 and 443, disabling all the others.

Regarding the "**DMZ Network**", the firewall allows only the ssh protocol coming from the "Client Network" and HTTP/HTTPS protocol for the "webserver", and outgoing is allowed to send logs, at the port 514, and use of DNS service by the "Domain Controller" and the Internet access. All of the other features are disallowed.

For "**Internal Server Network**", the internal firewall allows the log file coming from the other two LAN and the usage of the DNS service, moreover it allows the access via ssh protocol by the "Client Network".

In the end the "**Client Network**" can access ssh services, DNS, sending of the log file and the web service, all of other features are refused.

All the web and ssh services in the hosts are repeated between the IPv4 and Ipv6 addresses just duplicating the relative ip addresses.

Services of the ACME co.

Log File

To provide a functioning log server to store the log files created by the hosts of the network, we worked on the log server host in the server network. First of all we added the following in the /etc/rsyslog.conf file:

```
$template RemoteLogs, "/var/log/%HOSTNAME%/%HOSTNAME%.log"
```

```
*.* ?RemoteLogs
```

```
& ~
```

Thanks to this, whenever the log server receives a log input from an external host, it creates a .log file with the name of the source in the /var/log directory.

To make possible to receive the log file into the standard UDP port, we simply removed the "#" comment to the "#Provides UDP syslog reception".

Regarding the hosts configuration, we also modified the rsyslog.conf file, in order to send all the log also to the server address thanks to the following line:

```
*.* @100.100.1.3:514
```

Where the @ means that the log messages are sent via UDP protocol together with the ip address of the log server and the defined port in the rsyslog.conf of the log server.

Proxy

Regarding the Proxy server service, on the zentyal GUI of the proxy server in the DMZ, we enabled the fact that the interface eth0 would be considered as a WAN interface, and later we enabled the proxy service in the modules section.

In the hosts, to provide the fact that the HTTP/HTTPS had to pass through the proxy server instead to go directly to the the destination, we modified the /etc/environment adding the following:

```
http_proxy=http://100.100.6.3:3128
```

```
https_proxy=https://100.100.6.3:3128
```

Where the Ip address is the one of the proxy server and the port is the port defined in the zentyal interface.

Secure Shell

















































In order to provide a SSH connection from kali (.100) to the other hosts of the DMZ and SERVERS network, we added a new user on those hosts (the user “user” with password “password”).

Later on the server side we modified the /etc/ssh/sshd_config file disabling the PermitRootLogin and the X11Forwarding enabling the ssh connection just via username and password.













































Policy Implementation In Opnsense

MAIN FIREWALL

DMZ

	  	IPv4 UDP	*	*	100.100.1.3	514	*	*	Syslog	   
	  	IPv4 TCP/UDP	100.100.6.3	*	*	53 - 443	*	*	Proxy Internet Access	   
	  	IPv6 TCP/UDP	2001:470:b5b8:1e00:8410:9bff:fe35:525c	*	*	53 - 443	*	*	Proxy Internet Access (IPv6)	   
	  	IPv4 UDP	*	*	This Firewall	53 (DNS)	*	*	Allow DNS requests	   
	  	IPv4 *	*	*	*	*	*	*	Block any traffic in	   
	  	IPv6 *	*	*	*	*	*	*	Block any traffic in (IPv6)	   

EXTERNAL_CLIENTS

	  	IPv4 TCP/UDP	*	*	100.100.6.3	3128	*	*	allow access to proxy	   
	  	IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00:8410:9bff:fe35:525c/32	3128	*	*	Allow access to proxy (IPv6)	   
	  	IPv4 UDP	*	*	100.100.1.3	514	*	*	Syslog	   
	  	IPv6 *	*	*	*	*	*	*	Block traffic (IPv6)	   
	  	IPv4 *	*	*	*	*	*	*	Block traffic	   
										   

INTERNAL

<input type="checkbox"/>					IPv6 *	*	*	*	*	*	*	Allow all Ingoing				
<input type="checkbox"/>					IPv4 *	*	*	*	*	*	*	Allow all Ingoing				

WAN

<input type="checkbox"/>					IPv4 TCP	100.101.0.3/24	*	*		80 - 443	*	*	Connect from the host machine				
<input type="checkbox"/>					IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00:40fa:57ff:fe4a:2073		80 - 443	*	*	Web Service (IPv6)				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	100.100.6.2		80 - 443	*	*	Web Service				
<input type="checkbox"/>					IPv4 *	*	*	*		*	*	*	Block Traffic				
<input type="checkbox"/>					IPv6 *	*	*	*		*	*	*	Block Traffic (IPv6)				

INTERNAL FIREWALL





































CLIENT

<input type="checkbox"/>					IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00::/64		22 (SSH)	*	*	SSH for DMZ (IPv6)				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	100.100.6.0/24		22 (SSH)	*	*	SSH forDMZ				
<input type="checkbox"/>					IPv6 TCP/UDP	*	*	SERVERS net		22 (SSH)	*	*	SSH for Server Network (IPv6)				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	SERVERS net		22 (SSH)	*	*	SSH for Server Network				
<input type="checkbox"/>					IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00::/64		22 (SSH)	*	*	SSH for external client (IPv6)				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	100.100.4.0/24		22 (SSH)	*	*	SSH for external client				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	100.100.6.3		443 (HTTPS)	*	*	External Web Services				
<input type="checkbox"/>					IPv4 TCP/UDP	*	*	100.100.6.3		80 (HTTP)	*	*	External Web Services				
<input type="checkbox"/>					IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00:8410:9bff:fe35:525c		443 (HTTPS)	*	*	External Web Services (IPv6)				
<input type="checkbox"/>					IPv6 TCP/UDP	*	*	2001:470:b5b8:1e00:8410:9bff:fe35:525c		80 (HTTP)	*	*	External Web Services (IPv6)				
<input type="checkbox"/>					IPv4 TCP/UDP	CLIENTS net	*	100.100.6.3		3128	*	*	Allow Proxy connection				
<input type="checkbox"/>					IPv4 *	*	*	*		*	*	*	Block any traffic in				
<input type="checkbox"/>					IPv6 *	*	*	*		*	*	*	Block any traffic in				

EXTERNAL

<input type="checkbox"/>					IPv6 *	*	*	*	*	*	*	*	Allow All Ingoing				
<input type="checkbox"/>					IPv4 *	*	*	*	*	*	*	*	Allow All Ingoing				

SERVICES

<input type="checkbox"/>	 	IPv6 TCP/UDP	2001:470:b5b8:1ef0:4c34:16ff:fe3d:beb3	*	*	53 - 443	*	*	Allow outgoing packet for DC (IPv6)	   
<input type="checkbox"/>	 	IPv4 UDP	*	*	100.100.1.2	53 (DNS)	*	*	Allow DNS requests from MAIN FIREWALL	   
<input type="checkbox"/>	 	IPv4 TCP/UDP	100.100.1.2	*	*	53 - 443	*	*	Allow outgoing packet for DC	   
<input type="checkbox"/>	 	IPv4 TCP/UDP	*	*	100.100.6.3	3128	*	*	Allow proxy access	   
<input type="checkbox"/>	 	IPv4 *	*	*	*	*	*	*	Block any traffic in	   
<input type="checkbox"/>	 	IPv6 *	*	*	*	*	*	*	Block any traffic in	   

Test Of The Configuration

Firewall

Regarding the FIREWALL testing, we focused our attention on all the hosts of the network and test if they were working properly.

An example of the firewall test configuration, firstly we denied all the services and we see that the service didn't end correctly and then we enabled a correct rule in order to view the changes that the rules apply.

e.g. (Firstly block the port 514 to deny the Log services and see that no log file was received. After that we allow the use of 514 port in order to do the same test and see the correct execution)

DNS

As regard the **DNS**, we tested if all the hosts are correctly report their IP Address when we execute the following command: *host "host_name"*

e.g. (*host ws.giordele.com*)

e.g. (*host ws.giordele6.com*)

Also with the www.google.com host, external to the network.

SSH

As regard the **SSH**, we decided to test it thanks to the ssh command:

"ssh ssh_user@server_name", and see if the connection is established correctly. e.g.

(*ssh ssh_user@100.100.6.2*)

Web Server

As regard the **Web server**, we focus our attention on the correct execution of the http/https protocol in order to view the Web page.

e.g. (using client .100 browser "http://100.100.6.2")

Log File

As regard the **Log file**, we test if all the host send their logs to "logserver".

e.g. (while on the first machine execute some command like "/etc/init.d/networking restart", on the other machine we execute the command "less var/log/ClientName/ClientName.log", in order to view the log sent by the first host)

Final remarks

This is our proposed solution useful to configure IPv6 addressing, DNS services both for IPv4 and IPv6, security policies for the ACME.co services.

Regarding the IPv6 addressing we encountered some initial problems maybe because this was our first time that we managed this topic however this didn't stop us to find a good solution. We hope that firewall policies that we set up provide good security to the network.